

# CIAO : modèle de composants et framework OSGi pour des applications télécoms adaptables dynamiquement

Areski Flissi<sup>1</sup> and Gilles Vanwormhoudt<sup>1,2</sup>

<sup>1</sup> LIFL/CNRS - Université Lille 1 (UMR 8022)

<sup>2</sup> Institut TELECOM

59655 Villeneuve d'Ascq cedex - France  
{Areski.Flissi, Gilles.Vanwormhoudt}@lifl.fr

**Abstract.** Nous présentons CIAO (Components for sIp ApplicatiOns), un modèle de composants hiérarchique et dynamique, spécifique au domaine des services télécoms, ainsi que son implémentation sous la forme d'un framework d'exécution au dessus de la plate-forme OSGi. L'originalité de CIAO est qu'il permet de concevoir des applications télécoms avancées adaptables dynamiquement.

## 1 Introduction

Avec l'évolution rapide des réseaux IP d'une part, l'apparition et l'adoption de nouveaux protocoles tel SIP d'autre part, le domaine des services télécoms, et en particulier le développement d'applications avancées prenant en compte divers aspects tels la présence, la mobilité, la localisation, etc., nécessite aujourd'hui de nouvelles méthodologies et techniques inspirées du génie logiciel. Nous avons proposé dans [2], un modèle de programmation basé sur les notions d'acteurs, sessions et rôles afin de répondre aux différents challenges posés par la conception d'applications faisant intervenir plusieurs entités distribuées impliquées dans des interactions complexes. Dans [1], nous avons développé un outillage IDM, ainsi qu'un langage dédié basé sur ce modèle. Nous proposons ici de nous intéresser à l'évolution de ces applications, après le déploiement ou durant l'exécution, afin d'adapter leur comportement ou d'ajouter de nouvelles fonctionnalités dynamiquement. Dans ce poster, nous présentons CIAO (Components for sIp ApplicatiOns), un modèle de composant spécifique au domaine des applications télécoms répondant à cet objectif, ainsi que son implémentation sous la forme d'un framework d'exécution, reposant sur la plate-forme OSGi.

## 2 Modèle de composants CIAO

Nous avons défini un modèle de composants hiérarchique et dynamique, spécifique aux applications télécoms basées sur SIP. Celui-ci fait intervenir trois principaux types de composant qui sont:

- Un composant acteur nommé **Actor** qui représente une entité distribuée communiquant, via un flux de messages SIP, avec d'autres entités. Un acteur peut participer à différentes sessions (de même type ou non) avec d'autres acteurs. Un composant acteur est un composite encapsulant des composants **SessionPart** représentant chaque participation d'un acteur à une session. Le composite acteur a en charge l'aiguillage du flux de messages SIP au niveau des **SessionParts** et la coordination de ceux-ci (*i.e.* création d'une nouvelle instance, transmission d'un message à un composant **SessionPart** existant, etc.).
- Un composant **SessionPart** représente une participation d'un acteur à une session<sup>3</sup> donnée. **SessionPart** encapsule l'ensemble des comportements de l'acteur (*i.e.* ces rôles) au sein d'une session. Ainsi, un composant **SessionPart** est un composite contenant un ensemble de composants **Role**, dont il gère le cycle de vie.
- Un composant **Role**, qui est la brique de base du modèle contenant tout ou partie du comportement d'acteur relativement à une session. Ce comportement consiste à réaliser la logique métier en fonction de l'état et des messages SIP échangés.

---

<sup>3</sup> Nous entendons ici par session, un échange de messages persistant entre plusieurs acteurs

La particularité de ce modèle est la gestion dynamique des composants, basée sur les sessions SIP réelles. Les différents composants sont créés/détruits dynamiquement en fonction des flux de messages SIP, de l'état de l'acteur, des sessions, etc. Adapter dynamiquement une application en cours d'exécution consiste donc simplement à instancier ou détruire de nouveaux types de composants. Au sein d'un acteur, les composants interagissent pour router et traiter les messages SIP échangés, grâce à des ports spécifiques. Il existe également des moyens de coordination entre les composites et les sous-composants afin de supporter des comportements complexes impliquant plusieurs sessions d'un acteur ou plusieurs rôles d'une session.

### 3 CIAO : un framework au dessus de la plate-forme OSGi

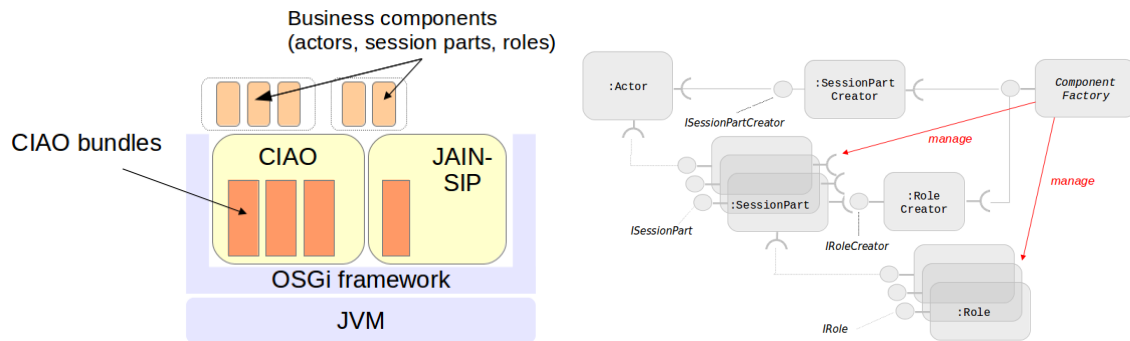


Fig. 1. Architecture et implémentation des composants CIAO

Le framework CIAO se base sur OSGi et l'architecture JAIN-SIP, comme illustré sur la figure 1 (partie gauche). Le choix d'OSGi a en partie été guidé par les fonctionnalités dynamiques proposées par cette spécification et plus particulièrement les composants *Declarative Service* (DS) présents depuis la version 4 d'OSGi, qui permettent de s'abstraire des problèmes de gestion dynamique des dépendances entre services requis et fournis des composants. Les composants CIAO ont donc été concrètement implémentés avec des composants DS. La partie droite de la figure 1 montre en détail l'assemblage des composants CIAO. La gestion de la dynamique, donc du cycle de vie des composants `SessionPart` et `Role` a été rendu possible grâce à l'utilisation du service *Component-Factory* offert par la plate-forme OSGi, par les composants `SessionPartCreator` et `RoleCreator`. CIAO permet l'ajout, le retrait et le remplacement des comportements d'acteurs relativement aux sessions. Ces adaptations peuvent être appliquées sans interruption de l'application, grâce à une gestion de vie du cycle des composants qui tient compte des sessions en cours.

### References

1. Areski Flissi and Gilles Vanwormhoudt. Programmation orientée domaine pour les services télécoms : concepts, DSL et outillage. In *Conférence en Ingénierie du Logiciel (CIEL 2012)*, pages 1–6, Rennes, France, 2012.
2. Gilles Vanwormhoudt and Areski Flissi. Session-based Role Programming for the Design of Advanced Telephony Applications. In *Proceedings of Distributed Applications and Interoperable Systems (DAIS'11)*, pages 77–91, 2011.